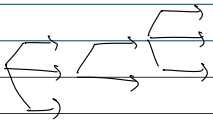
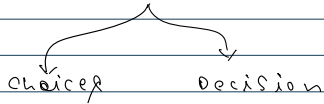


What is Recursion?

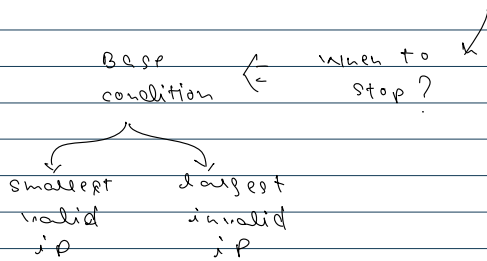
It is a method in which a fⁿ calls itself usually with smaller inputs.

```
A() {
    A()
    B();
}
```

Where do we need recursion?



$f(n) \rightarrow f(n-1) \rightarrow f(n-2) \rightarrow f(n-3) \dots$



"Recursion works like magic"

$f(n) \rightarrow f(n-1)$ ↗ Trust that $f(n-1)$ will always return a valid o/p.

Print numbers from 1 to n

$n = 5$
 1
 2
 3
 4
 5
 i/p

print(5)

1 2 3 4 5 ← print(4)

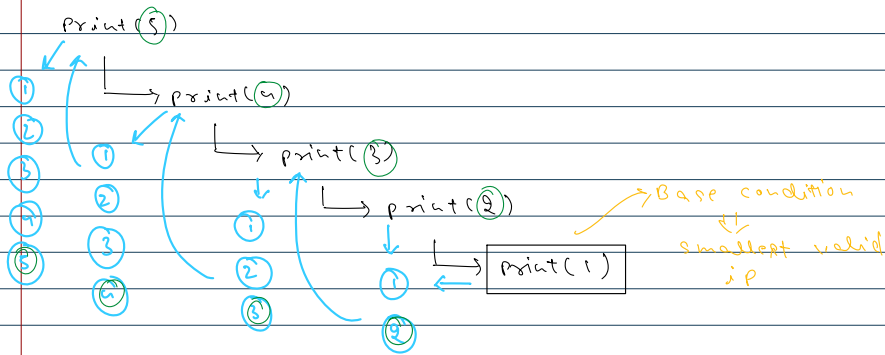
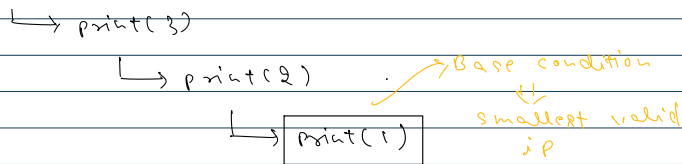
print(5)

→ print(4)

→ print(3)

→ print(2)

↗ Base condition

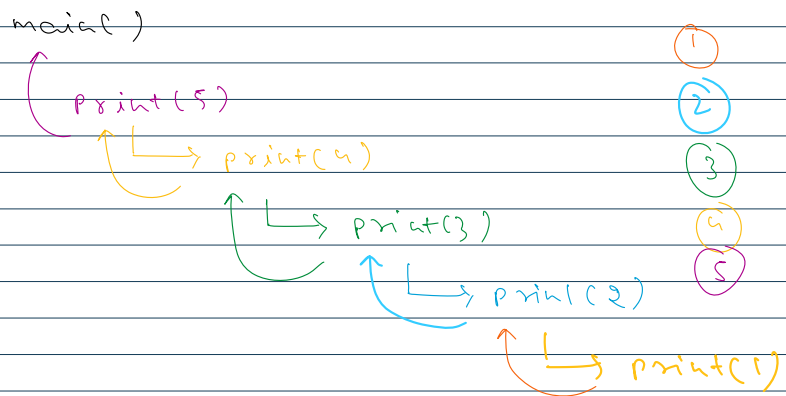


```

public class Main {
    public static void main(String[] args) {
        ✓ int n = 5;
        ✓ print(n);
    }

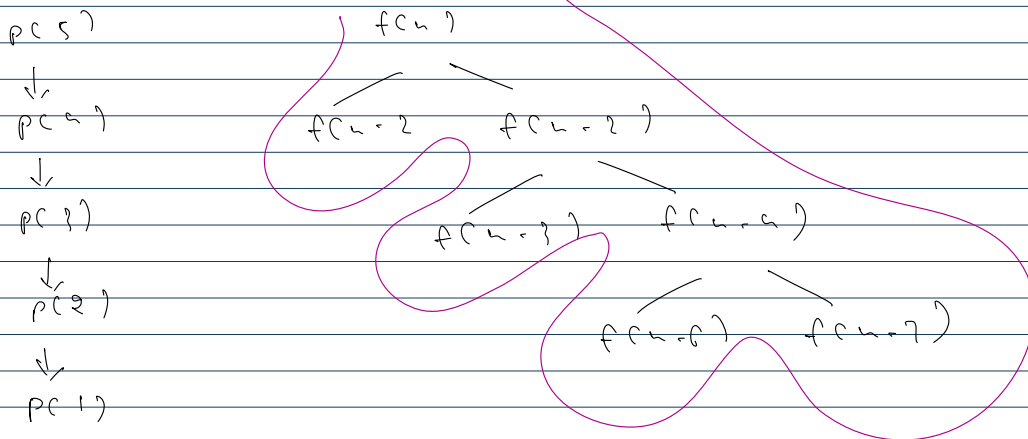
    public static void print(int n){
        if(n==1){
            System.out.println(n);
            return;
        }
        print(n-1);
        System.out.println(n);
    }
}

```

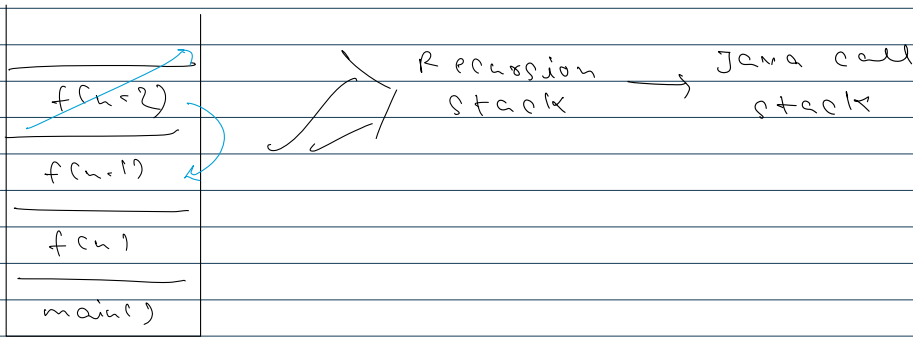


Recursive tree

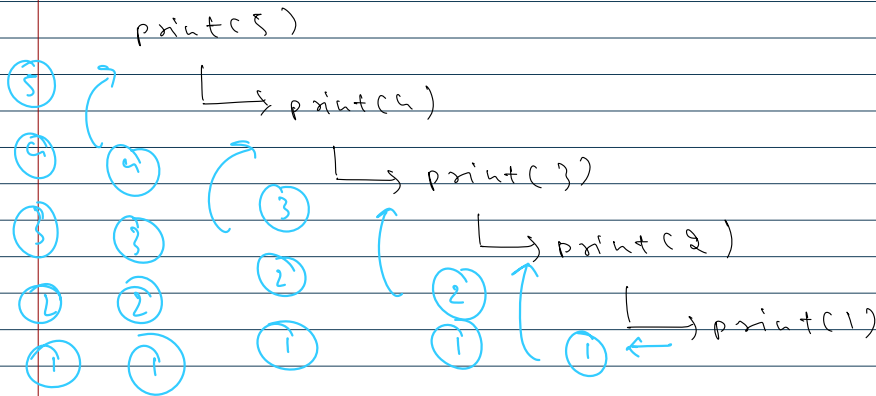
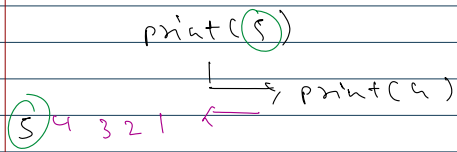
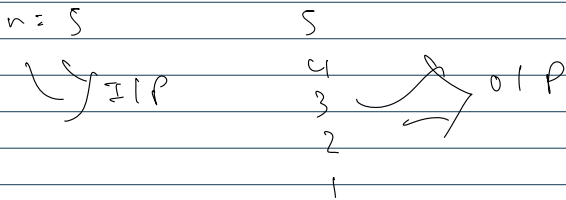
It visualises the code flow of a recursive code.



$fcn() \rightarrow fcn-2() \rightarrow fcn-2()$



Print numbers from n to 1



```
public class Main {
    public static void main(String[] args) {
        int n = 5;
        printDecreasing(n);
    }

    public static void printDecreasing(int n){
        if(n==1){
            System.out.println(n);
            return;
        }

        System.out.println(n);
        printDecreasing(n-1);
    }
}
```

Factorial of a number

$$5! = 5 \times 4 \times 3 \times 2 \times 1$$

$$5! = 5 \times 4!$$

$$\text{fact}(n) = n \times \text{fact}(n-1)$$

$$\text{fact}(5) = 5 \times \text{fact}(4)$$

Base condition

↓

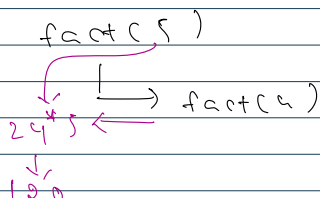
Smallest valid i/p

↓

n = 0 or 1

↓

return 1

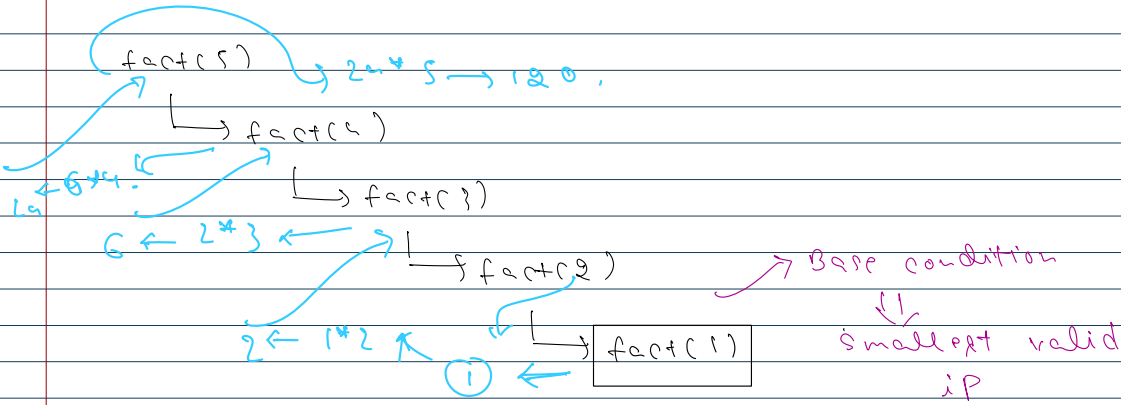


FACTUAL

120

```
public class Main {
    public static void main(String[] args) {
        int n = 5;
        System.out.println(fact(n));
    }

    public static int fact(int n){
        if(n==1){
            return 1;
        }
        return n*fact(n-1);
    }
}
```



Power a number

$3^3 \rightarrow 27$

$\text{pow}(\text{base}, \text{exp}) \rightarrow ?$

$\text{pow}(3, 2) \rightarrow 9 \text{ i.e. } 3^2$

$\text{pow}(a, n)$

$3^4 \rightarrow 3 \cdot 3^3$

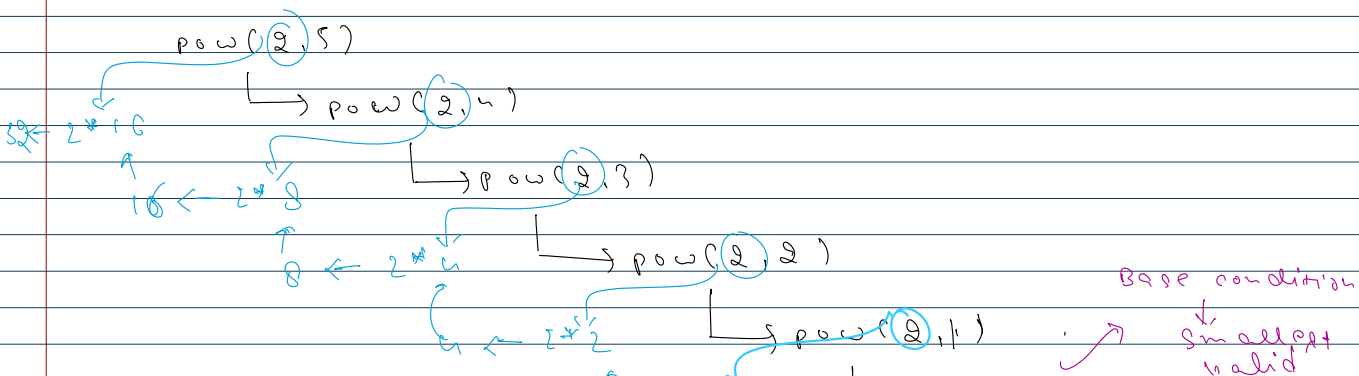
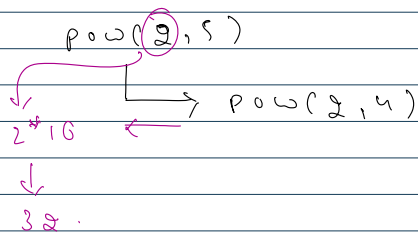
$3^n \rightarrow 3 \cdot 3^{n-1}$

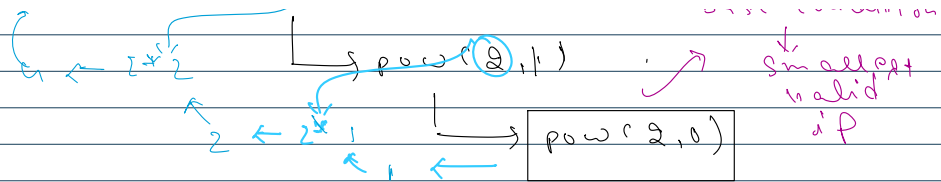
$\text{pow}(a, n) \rightarrow a * \text{pow}(a, n-1)$

$a^5 \rightarrow a * a^4$

$a^n \rightarrow a \cdot a^{n-1}$

Base condition $\Rightarrow n == 0 \Rightarrow \text{return } 1 \Rightarrow \text{smallest valid ip.}$





```
public class Main {
    public static void main(String[] args) {
        int a = 2;
        int n = 5;
        System.out.println(pow(a, n));
    }

    public static int pow(int a, int n){
        if(n==0){
            return 1;
        }
        return a*pow(a, n-1);
    }
}
```

First occurrence of a number in an array

arr → 1, 2, 5, 4, 3, 4, 7, 4, 3, 6
 item → 4
 o/p → 3

```
public class Main {
    public static void main(String[] args) {
        int[] arr = {1, 2, 5, 4, 3, 7, 8, 1, 6, 8};
        int item = 44;
        int index = 0;
        System.out.println(firstOccurrence(arr, item, index));
    }

    public static int firstOccurrence(int[] arr, int item, int index){
        if(index == arr.length){
            return -1;
        }

        if(arr[index] == item){
            return index;
        }
        return firstOccurrence(arr, item, index+1);
    }
}
```

Check if a array is sorted or not

arr → { 5, 7, 8, 11, 22, 27 } ; ∴, ascending order
 o/p → true

$arr[i] > arr[i+1] \Rightarrow \text{false}.$

```
public class Main {
    public static void main(String[] args) {
        int[] arr = {1, 2, 3, 4}; //{1, 2, 5, 4, 3, 7, 8, 1, 6, 8};
        int index = 0;
        System.out.println(isSorted(arr, index));
    }

    public static boolean isSorted(int[] arr, int index){
        if(index == arr.length-1){
            return true;
        }

        if(arr[index] > arr[index+1]){
            return false;
        }
        return isSorted(arr, index+1);
    }
}
```

Reverse words in a string

```
public class Main {
    public static void main(String[] args) {
        String s = "the sky is blue";
        String[] arr = s.split(" ");

        for(String str: arr){
            System.out.println(str);
        }

        String a = " abtgyj ";
        System.out.println(a);
        a = a.trim();
        System.out.println(a);
    }
}
```

```
class Solution {
    public String reverseWords(String s) {
        s = s.trim();
        String[] arr = s.split(" ");
        String op = "";
        for(int i=arr.length-1; i>=0; i--){
            op = op + arr[i] + " ";
        }
        return op.trim();
    }
}
```